WHAT IS CLAIMED IS:

1. A method for updating data in a storage device, comprising:

receiving an update to one or more blocks of customer data at addresses in the storage device;

for each block of data to update, generating metadata indicating the address of the block in the storage device and an error checking code that is capable of being used to determine whether the customer data in the block has changed;

for each block of data to update, writing the block of data to update and the metadata for the block to cache; and

for each block of data to update, transferring the block of data and the metadata for the block from the cache to the storage device.

2. The method of claim 1, further comprising

for each block of data to update, determining whether the address of the block of data in the metadata and the address in the storage device to update match;

for each block of data to update, performing an operation on the customer data in the block and the error checking code to determine whether the customer data has changed, wherein the block of data to update and metadata for the block is transferred to the storage device if the address of the block in the metadata and requested address match and the customer data has not changed.

3. The method of claim 2, wherein the operation performed on the customer data comprises XORing the customer data and wherein the error checking code comprises a longitudinal redundancy checking code.

4. The method of claim 1, further comprising:

setting up a control block including the address of a first block of data to update in the storage device and an instruction to generate the address and error code as metadata

for the block, wherein generating the metadata indicating the address of the block in the storage device comprises using the block address in the control block as the address of the block in the storage device to write as the metadata.

5. The method of claim 4, further comprising:

for each block to update, incrementing the block address in the control block before processing a next block, wherein the incremented block address is used as the address in the storage device to generate as metadata for the next block of data to update.

6. The method of claim 1, wherein generating the error checking code comprises XORing the customer data in the block such that the error checking code comprises a longitudinal redundancy checking code.

7. The method of claim 2, wherein the steps of generating the metadata and determining whether the address of the block in the storage device and block address in metadata match and performing the operation on the customer data and error checking is performed by a device that is separate from a main processor, wherein the device transfers the block of data from the cache to the storage device using a direct memory access (DMA) channel.

8. The method of claim 2, further comprising:

setting up a control block in the cache including the block address of a first block to update and an instruction to check the address and error code of the block, wherein determining whether the block address in the metadata in the storage device and the block address to update match comprises using the block address in the control block in the cache as the block address to update to compare with the block address in the metadata stored with the block in the cache.

Express Mail No. EL484106684US
Docket No. TUC9-2000-0013US1
Firm No. 0018.0074

9. The method of claim 8, further comprising:

for each requested block, incrementing the block address in the first control block after transferring the block from the cache to the storage device, wherein the incremented block address is used as the block address to compare with the block address in the metadata in the cache for the next requested block.

10. The method of claim 1, further comprising:

recovering from a power loss; and

using the metadata for blocks in cache to rebuild cache control blocks for the blocks in cache after recovering from the power loss.

11. The method of claim 10, wherein the blocks of data and metadata are stored in a non-volatile portion of the cache and the cache control blocks are stored in a volatile portion of the cache.

12. The method of claim 10, further comprising:

maintaining a first data structure indicating whether each block of data in cache is valid or invalid and a second data structure indicating whether each block of data includes modified or unmodified data;

for each block of data in the cache, using the error checking code to determine whether the block of data in the cache has changed; and

for each block of data in the cache, indicating in the first data structure that the block of data is invalid if the second data structure indicates that the block of data is not modified and the block of data has changed.

13. The method of claim 10, further comprising:

maintaining a first data structure indicating whether each block of data in cache is valid or invalid and a second data structure indicating whether each block of data includes modified or unmodified data;

for each block of data in the cache, using the error checking code to determine whether the block of data in the cache has changed; and

for each block of data in the cache accessing the update to the block from another storage location if the second data structure indicates that the block of data is modified and the block of data has changed.

14. The method of claim 10, further comprising:

maintaining a first data structure indicating whether each block of data in cache is valid or invalid and a second data structure indicating whether each block of data includes modified or unmodified data;

for each block of data in the cache, using the error checking code to determine whether the block of data in the cache has changed; and

rebuilding the cache control block for the block using the address information in the metadata for the block if the block of data has not changed.

15. The method of claim 1, wherein the error checking code is further capable of being used to determine whether the metadata in the block has changed.

16. A system for updating data, comprising:

a storage device,

means for receiving an update to one or more blocks of customer data at addresses in the storage device;

means for generating metadata, for each block of data to update, indicating the address of the block in the storage device and an error checking code that is capable of being used to determine whether the customer data in the block has changed;

means for writing the block of data to update and the metadata for the block to cache for each block of data to update; and

means transferring the block of data and the metadata for the block from the cache to the storage device for each block of data to update.

17. The system of claim 16, further comprising

means for determining whether the address of the block of data in the metadata and the address in the storage device to update match for each block of data to update;

means for performing an operation on the customer data in the block and the error checking code to determine whether the customer data has changed for each block of data to update, wherein the block of data to update and metadata for the block is transferred to the storage device if the address of the block in the metadata and requested address match and the customer data has not changed.

18. The system of claim 17, wherein the operation performed on the customer data comprises XORing the customer data and wherein the error checking code comprises a longitudinal redundancy checking code.

19. The system of claim 16, further comprising:

means for setting up a control block including the address of a first block of data to update in the storage device and an instruction to generate the address and error code as metadata for the block, wherein the means for generating the metadata indicating the address of the block in the storage device comprises using the block address in the control block as the address of the block in the storage device to write as the metadata.

Express Mail No. EL484106684US
Docket No. TUC9-2000-0013US1
Firm No. 0018.0074

20. The system of claim 19, further comprising:

means for incrementing the block address in the control block, for each block to update, before processing a next block, wherein the incremented block address is used as the address in the storage device to generate as metadata for the next block of data to update.

21. The system of claim 16, wherein the means for generating the error checking code comprises XORing the customer data in the block such that the error checking code comprises a longitudinal redundancy checking code.

22. The system of claim 17, wherein the means for generating the metadata, determining whether the address of the block in the storage device and block address in metadata match, and performing the operation on the customer data and error checking comprises a device that is separate from a main processor, wherein the device transfers the block of data from the cache to the storage device using a direct memory access (DMA) channel.

23. The system of claim 17, further comprising:

means for setting up a control block in the cache including the block address of a first block to update and an instruction to check the address and error code of the block, wherein the means for determining whether the block address in the metadata in the storage device and the block address to update match comprises using the block address in the control block in the cache as the block address to update to compare with the block address in the metadata stored with the block in the cache.

24. The system of claim 23, further comprising:

means for incrementing the block address in the first control block, for each requested block, after transferring the block from the cache to the storage device, wherein the incremented block address is used as the block address to compare with the block address in the metadata in the cache for the next requested block.

R1.126

~~24.~~ 25. The system of claim 16, further comprising:

means for recovering from a power loss; and

means for using the metadata for blocks in cache to rebuild cache control blocks for the blocks in cache after recovering from the power loss.

R1.126

~~25.~~ 26. The system of claim 24, wherein the blocks of data and metadata are stored in a non-volatile portion of the cache and the cache control blocks are stored in a volatile portion of the cache.

R1.126

~~26.~~ 27. The system of claim 24, further comprising:

means for maintaining a first data structure indicating whether each block of data in cache is valid or invalid and a second data structure indicating whether each block of data includes modified or unmodified data;

means for using the error checking code, for each block of data in the cache, to determine whether the block of data in the cache has changed; and

means for indicating in the first data structure, for each block of data in the cache, that the block of data is invalid if the second data structure indicates that the block of data is not modified and the block of data has changed.

R1.126

28. 27. The system of claim 24, further comprising:

means for maintaining a first data structure indicating whether each block of data in cache is valid or invalid and a second data structure indicating whether each block of data includes modified or unmodified data;

means for using the error checking code, for each block of data in the cache, to determine whether the block of data in the cache has changed; and

means for accessing, for each block of data in the cache, the update to the block from another storage location if the second data structure indicates that the block of data is modified and the block of data has changed.

R1.126

29. 28. The system of claim 24, further comprising:

means for maintaining a first data structure indicating whether each block of data in cache is valid or invalid and a second data structure indicating whether each block of data includes modified or unmodified data;

means for using the error checking code to determine whether the block of data in the cache has changed for each block of data in the cache; and

means for rebuilding the cache control block for the block using the address information in the metadata for the block if the block of data has not changed.

R1.126

30. 29. The system of claim 16, further comprising means for using the error checking code to determine whether the metadata in the block has changed.

R1.126

31. 30. An integrated circuit device including logic for updating data in a storage device, wherein the logic performs:

receiving an update to one or more blocks of customer data at addresses in the storage device;

for each block of data to update, generating metadata indicating the address of the block in the storage device and an error checking code that is capable of being used to determine whether the customer data in the block has changed;

for each block of data to update, writing the block of data to update and the metadata for the block to cache; and

for each block of data to update, transferring the block of data and the metadata for the block from the cache to the storage device.

R1.126

32. ~~31.~~ The integrated circuit device of claim 30, wherein the logic further performs:

for each block of data to update, determining whether the address of the block of data in the metadata and the address in the storage device to update match;

for each block of data to update, performing an operation on the customer data in the block and the error checking code to determine whether the customer data has changed, wherein the block of data to update and metadata for the block is transferred to the storage device if the address of the block in the metadata and requested address match and the customer data has not changed.

R1.126

33. ~~32.~~ The integrated circuit device of claim 31, wherein the operation performed on the customer data comprises XORing the customer data and wherein the error checking code comprises a longitudinal redundancy checking code.

R1.126

34. ~~33.~~ The integrated circuit device of claim 30, wherein a processor generates a control block including the address of a first block of data to update in the storage device and an instruction to cause the integrated circuit device to generate the address and error code as metadata for the block, and wherein the integrated circuit device generates the metadata indicating the address of the block in the storage device by using the block

address in the control block as the address of the block in the storage device to write as the metadata.

R1.126 35. ~~34.~~ The integrated circuit device of claim 33, wherein the logic further performs:

for each block to update, incrementing the block address in the control block before processing a next block, wherein the incremented block address is used as the address in the storage device to generate as metadata for the next block of data to update.

R1.126 36. ~~35.~~ The integrated circuit device of claim 30, wherein generating the error checking code comprises XORing the customer data in the block such that the error checking code comprises a longitudinal redundancy checking code.

R1.126 37. ~~36.~~ The integrated circuit device of claim 31, wherein the integrated circuit device is included in a storage system having a main processor, wherein the device transfers the block of data from the cache to the storage device using a direct memory access (DMA) channel.

R1.126 38. ~~37.~~ The integrated circuit device of claim 31, wherein a processor generates a control block in the cache including the block address of a first block to update and an instruction to cause the integrated circuit device to check the address and error code of the block, wherein the integrated circuit device determines whether the block address in the metadata in the storage device and the block address to update match by using the block address in the control block in the cache as the block address to update to compare with the block address in the metadata stored with the block in the cache.

Express Mail No. EL484106684US
Docket No. TUC9-2000-0013US1
Firm No. 0018.0074

R1.126 39. 38. The integrated circuit device of claim 37, wherein the logic further performs:

for each requested block, incrementing the block address in the first control block after transferring the block from the cache to the storage device, wherein the incremented block address is used as the block address to compare with the block address in the metadata in the cache for the next requested block.

R1.126 40. 39. The integrated circuit device of claim 30, wherein the logic further performs:

recovering from a power loss; and

using the metadata for blocks in cache to rebuild cache control blocks for the blocks in cache after recovering from the power loss.

R1.126 41. 40. The integrated circuit device of claim 39, wherein the blocks of data and metadata are stored in a non-volatile portion of the cache and the cache control blocks are stored in a volatile portion of the cache.

R1.126 42. 41. The integrated circuit device of claim 30, wherein the error checking code is further capable of being used to determine whether the metadata in the block has changed.

R1.126 43. 42. A computer readable medium including at least one data structure used for updating data in a storage device, comprising:

blocks of customer data;

a block of metadata for each block of customer data, wherein the metadata includes the address of the block in the storage device and an error checking code

that is capable of being used to determine whether the customer data in the block has changed while the block is in the cache, wherein block of data to update and the metadata for the block are written to cache.

R1.126 ~~43.~~ 44. The computer readable medium of claim 42, further comprising:

a control block including the address of a first block of data to update in the storage device and an instruction to generate the address and error code as metadata for the block, wherein generating the metadata indicating the address of the block in the storage device comprises using the block address in the control block as the address of the block in the storage device to write as the metadata.

R1.126 ~~44.~~ 45. The computer readable medium of claim 43, wherein the block address in the control block is incremented before processing a next block, wherein the incremented block address is used as the address in the storage device to generate as metadata for the next block of data to update.

R1.126 ~~45.~~ 46. The computer readable medium of claim 42, further comprising:

a control block in the cache including the block address of a first block to update and an instruction to check the address and error code of the block, wherein determining whether the block address in the metadata in the storage device and the block address to update match comprises using the block address in the control block in the cache as the block address to update to compare with the block address in the metadata stored with the block in the cache.

Express Mail No. EL484106684US
Docket No. TUC9-2000-0013US1
Firm No. 0018.0074

R1.126

47. ~~46.~~ The computer readable medium of claim 42, wherein the blocks of data and metadata are stored in a non-volatile portion of the cache and the cache control blocks are stored in a volatile portion of the cache.

R1.126

48. ~~47.~~ The computer readable medium of claim 42, further comprising:

a first data structure indicating whether each block of data in cache is valid or invalid and a second data structure indicating whether each block of data includes modified or unmodified data, wherein the error checking code is used to determine whether each block of data in the cache has changed in the event of a data recovery event, wherein the first data structure is modified to indicate that a block of data is invalid if the second data structure indicates that the block of data is not modified and the block of data has changed.

and A2